
Saving Wildlife with Generative AI: Latent Composite Flow Matching for Poaching Prediction

Lingkai Kong^{1*} Haichuan Wang^{1*} Charles A. Emogor^{1,2} Vincent Börsch-Supan¹
Lily Xu³ Milind Tambe¹

¹ School of Engineering and Applied Sciences, Harvard University

²Department of Computer Science and Technology, University of Cambridge

³Department of Industrial Engineering and Operations Research, Columbia University

Abstract

Poaching poses significant threats to wildlife and biodiversity. A valuable step in reducing poaching is to forecast poacher behavior, which can inform patrol planning and other conservation interventions. Existing poaching prediction methods based on linear models or decision trees lack the expressivity to capture complex, nonlinear spatiotemporal patterns. Recent advances in generative modeling, particularly flow matching, offer a more flexible alternative. However, training such models on real-world poaching data faces two central obstacles: imperfect detection of poaching events and limited data. To address imperfect detection, we integrate flow matching with an occupancy-based detection model and train the flow in latent space to infer the underlying occupancy state. To mitigate data scarcity, we adopt a composite flow initialized from a linear-model prediction rather than random noise which is the standard in diffusion models, injecting prior knowledge and improving generalization. Evaluations on datasets from two national parks in Uganda show consistent gains in predictive accuracy.

1 Introduction

Biodiversity loss is not only a conservation crisis but also a climate challenge: the decline of keystone species disrupts ecosystems, diminishes carbon storage, and weakens resilience to climate change [1]. Wildlife poaching is a major driver of this loss. Elephants, pangolins, and lions face intense hunting pressure—tens of thousands are killed annually for ivory, scales, or body parts—threatening both species survival and ecosystem stability [2–4]. A central challenge is how to allocate limited patrol resources to the locations where they can most effectively deter poaching.

Machine learning has been applied to forecast poaching risk and guide patrol planning. However, existing models—such as linear and tree-based approaches [5–7]—struggle to capture complex spatial dependencies and poachers’ adaptive strategies. They often make simplifying independence assumptions, leading to inaccurate risk maps and suboptimal patrol deployment. Recent advances in generative AI, such as diffusion models [8] and flow matching [9], provide powerful tools for modeling high-dimensional and strategic behavior. Yet applying these methods to poaching data presents two key challenges: (i) **data scarcity**,



Figure 1: Snares and rangers conducting a patrol. Photo: Uganda Wildlife Authority.

*Equal contribution

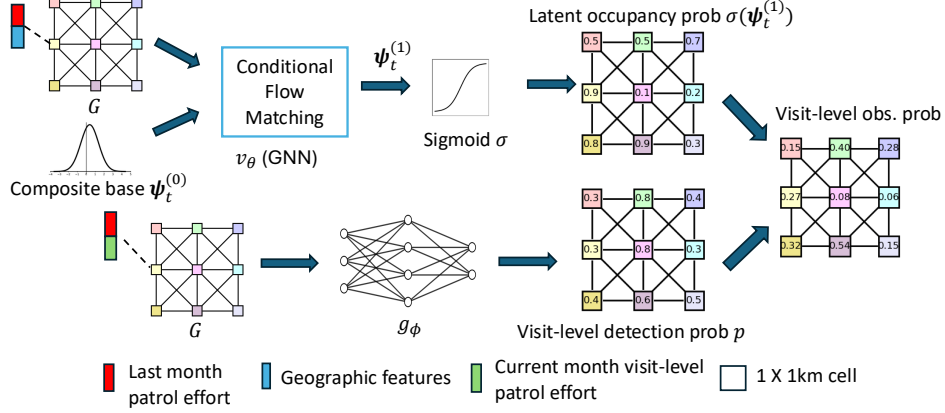


Figure 2: Overview of the latent composite flow framework. **Upper branch (latent generation):** 1×1 km cells are nodes in a graph G with edges between adjacent cells. For month t , node features comprise geospatial covariates $\mathbf{x}_{i,t}$ and last month’s aggregated patrol effort $a_{i,t-1}^m = \sum_j a_{i,t-1,j}$. A *composite base* initializes the latent logits $\psi_t^{(0)}$ from a pretrained linear occupancy model with Gaussian perturbation. A graph-conditional velocity field v_θ transports $\psi_t^{(0)}$ to $\psi_t^{(1)}$ via conditional flow matching; $\sigma(\psi_t^{(1)})$ yields occupancy probabilities. **Lower branch (detection):** the visit-level detector $g_\phi(\mathbf{x}_{i,t}, a_{i,t,j})$ uses geospatial features and current-month visit effort to produce detection probabilities $\{p_{i,t,j}\}$, which are combined with occupancy in the occupancy–detection likelihood.

since historical records are limited in many parks, and (ii) **imperfect detection**, as many poaching incidents remain unseen due to hidden snares or dense vegetation [10].

We address these challenges with WILDFLOW, a latent composite flow matching framework. Building on ecological occupancy models [11], we represent poaching activity as a latent state separate from imperfect detection, enabling explicit handling of missing signals. To mitigate data scarcity, we introduce a composite flow that initializes the generative process from a linear model’s predictions rather than random noise, embedding domain knowledge to improve generalization. On real-world datasets from two Ugandan national parks, WILDFLOW improves AUPR by 7–10% over baselines demonstrating the potential of generative AI to strengthen conservation and climate resilience.

2 Related Works

Early work in conservation security used behavioral models such as SUQR in PAWS [12]. Imperfect detection was later addressed via Bayesian networks with latent variables [13], though these proved brittle in practice, where decision-tree ensembles were more accurate and efficient [7]. Hybrid approaches combined geo-clustering with ensembles of decision trees [14], while stratified ensembles of weak learners (trees, Gaussian processes) tackled uncertainty from uneven patrol effort and were deployed in SMART [15–17]. Logistic regression has also been used to quantify deterrence effects of ranger patrols [5]. Despite these advances, most approaches rely on classical models that struggle with high-dimensional, nonlinear spatial patterns, and few exploit modern deep learning.

3 Background

3.1 Problem Definition

We study two Ugandan parks: Murchison Falls (MFNP, $\sim 5,000 \text{ km}^2$) and Queen Elizabeth (QENP, $\sim 2,500 \text{ km}^2$), both critical for biodiversity, carbon storage, and ecotourism [18]. Each park is partitioned into 1×1 km cells $i \in \{1, \dots, N\}$ and months $t \in \{1, \dots, T\}$. For each cell–month, features $\mathbf{x}_{i,t}$ include static geospatial attributes (e.g., elevation, distance to rivers) and dynamic covariates (e.g., precipitation, temperature, productivity). The latent occupancy state $z_{i,t} \in \{0, 1\}$ indicates whether poaching occurs. Rangers conduct GPS-tracked patrols and record observations. In month t , cell i may be visited $J_{i,t}$ times, with patrol effort $a_{i,t,j}$ and binary detection outcome $y_{i,t,j}$.

We aim to forecast poaching risk $p(z_{i,t} = 1)$ using features $\{\mathbf{x}_{i,t}\}$ and past effort $\{a_{i,t-1,j}\}$, while explicitly modeling visit-level effort $\{a_{i,t,j}\}$ and imperfect detection.

3.2 Preliminaries on Flow Matching

To capture complex spatial structure, we adopt *flow matching* (FM), a generative framework that learns a time-dependent velocity field to transport a base distribution toward the data distribution [9]. FM enables faster sampling than diffusion models [8], while often achieving comparable quality.

In this work, we focus on the *conditional* version. At flow time $s=0^2$, FM draws a base sample $\psi^{(0)} \sim p_0(\cdot | \mathbf{c})$ and evolves it under a conditional velocity field $\frac{d\psi^{(s)}}{ds} = v_\theta(\psi^{(s)}, s; \mathbf{c})$, where $\mathbf{c}$ denotes contextual information. Integrating from $s=0$ to $s=1$ yields $\psi^{(1)}$, a sample from $p_{\text{data}}(\cdot | \mathbf{c})$.

While FM can capture complex spatial patterns, it does not by itself address (i) imperfect detection and (ii) small data in conservation.

4 Proposed Method

We introduce WILDFLOW, a latent composite flow-matching framework that captures high-dimensional spatial patterns in poaching prediction while addressing imperfect detection and data scarcity. Figure 2 shows the overview of the data generation process.

Imperfect detection. Following ecological occupancy models [11], we represent poaching activity as a latent state $z_{i,t} \in \{0, 1\}$, distinct from visit-level detection outcomes $y_{i,t,j}$. Given patrol effort $a_{i,t,j}$, detection probability is $p_{i,t,j} = \sigma(g_\phi(\mathbf{x}_{i,t}, a_{i,t,j}))$, and the occupancy-detection likelihood is

$$\mathcal{L}_{\text{occ}}(i, t) = \log \left[(1 - r_{i,t})(1 - S_{i,t}) + r_{i,t} \prod_j p_{i,t,j}^{y_{i,t,j}} (1 - p_{i,t,j})^{1-y_{i,t,j}} \right], \quad (1)$$

where $r_{i,t} = \sigma(\psi_t^{(1)})$ is the latent occupancy probability and $S_{i,t}$ indicates at least one detection. To model the latent occupancy logits, we adopt conditional flow matching:

$$\frac{d\psi_t^{(s)}}{ds} = v_\theta(\psi_t^{(s)}, s; \mathbf{C}_t), \quad s \in [0, 1],$$

where $\mathbf{C}_t$ denotes cell features and past patrol effort, and v_θ is a graph-based velocity field propagating spatial context.

Data scarcity. To improve data efficiency, we depart from the standard Gaussian initialization used in existing flow matching and diffusion models, and instead warm-start the flow from a *composite base*: $\psi_t^{(0)} \sim \mathcal{N}(b_\eta(\mathbf{C}_t), \sigma_0^2 \mathbf{I})$, where b_η is a linear occupancy predictor and $\mathbf{C}_t$ denotes covariates and past patrol effort. This initialization injects domain priors while still allowing stochastic variation, leading to more data-efficient learning.

Training We adopt a two-stage procedure: (i) an encoder f_ω maps monthly observations, environmental features, and patrol effort into latent logits $\hat{\psi}_t^{(1)}$, which are optimized jointly with the detection head g_ϕ by maximizing $\sum_{i,t} \mathcal{L}_{\text{occ}}(i, t)$; (ii) we then freeze (f_ω, g_ϕ) and train the flow model v_θ to transport samples from the composite base $\psi_t^{(0)}$ toward the encoder’s surrogate logits $\hat{\psi}_t^{(1)}$ via squared-error flow matching.

5 Experiments

5.1 Setup

Data. We use ranger patrol and poaching records from MFNP (2014–2021) and QENP (2011–2016), provided by the Uganda Wildlife Authority. Following [19], we focus on historically high-risk subregions by selecting the 20 most active cells each month and expanding to groups of ≤ 25 adjacent cells. To handle non-stationarity, training for each test year uses the preceding three years of data [16].

Baselines. All methods share the same linear detection head. For latent occupancy we compare: Logistic Regression (LogReg), Gaussian Process (GP), MLP, Graph Neural Network (GNN), Transformer, Diffusion model, and our method. Full details are in Appendix A.4.

Metric. Since true poaching states are unobserved, we evaluate detection-level outcomes. For each cell-month, we compute the probability of at least one detection and compare against observed monthly labels using area under the precision–recall curve (AUPR).

²Throughout, t indexes months in the poaching data, while $s \in [0, 1]$ parameterizes the generative flow.

Park	Year	LogReg	GP	MLP	GNN	Transformer	Diffusion	Ours
MFNP	2017	0.305	0.278 $\pm$ 0.001	0.288 $\pm$ 0.017	0.336 $\pm$ 0.004	0.314 $\pm$ 0.016	0.313 $\pm$ 0.021	0.374 $\pm$ 0.032
	2018	0.344	0.308 $\pm$ 0.002	0.325 $\pm$ 0.004	0.260 $\pm$ 0.000	0.302 $\pm$ 0.064	0.279 $\pm$ 0.001	0.377 $\pm$ 0.011
	2019	0.406	0.359 $\pm$ 0.002	0.388 $\pm$ 0.007	0.526 $\pm$ 0.015	0.475 $\pm$ 0.141	0.362 $\pm$ 0.003	0.409 $\pm$ 0.009
	2020	0.421	0.401 $\pm$ 0.003	0.408 $\pm$ 0.004	0.366 $\pm$ 0.014	0.327 $\pm$ 0.002	0.439 $\pm$ 0.023	0.473 $\pm$ 0.006
	2021	0.360	0.364 $\pm$ 0.011	0.335 $\pm$ 0.010	0.430 $\pm$ 0.023	0.295 $\pm$ 0.085	0.342 $\pm$ 0.032	0.423 $\pm$ 0.004
	Avg	0.367	0.342	0.349	0.384	0.343	0.347	0.411
QENP	2014	0.119	0.107 $\pm$ 0.003	0.103 $\pm$ 0.003	0.136 $\pm$ 0.013	0.100 $\pm$ 0.002	0.102 $\pm$ 0.003	0.116 $\pm$ 0.009
	2015	0.180	0.156 $\pm$ 0.002	0.172 $\pm$ 0.006	0.111 $\pm$ 0.000	0.173 $\pm$ 0.016	0.162 $\pm$ 0.028	0.201 $\pm$ 0.017
	2016	0.258	0.220 $\pm$ 0.005	0.227 $\pm$ 0.004	0.238 $\pm$ 0.009	0.220 $\pm$ 0.019	0.201 $\pm$ 0.034	0.299 $\pm$ 0.027
	Avg	0.186	0.161	0.167	0.162	0.164	0.155	0.205

Table 1: AUPR comparison. All methods use a linear detection head for a fair comparison. Cells in green indicate a clear winner, while yellow denotes a practical tie.

Year	w/o base	w/o det.	Ours
2017	0.341 $\pm$ 0.008	0.278 $\pm$ 0.011	0.374 $\pm$ 0.032
2018	0.368 $\pm$ 0.019	0.265 $\pm$ 0.006	0.377 $\pm$ 0.011
2019	0.382 $\pm$ 0.006	0.301 $\pm$ 0.006	0.409 $\pm$ 0.009
2020	0.429 $\pm$ 0.007	0.350 $\pm$ 0.005	0.473 $\pm$ 0.006
2021	0.338 $\pm$ 0.040	0.290 $\pm$ 0.009	0.423 $\pm$ 0.004

Table 2: Ablation on MFNP (AUPR).

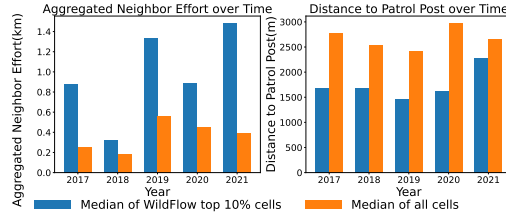


Figure 3: Case study on MFNP.

5.2 Results

Table 1 reports results on MFNP and QENP. We summarize our findings as follows. (1) WILDFLOW consistently outperforms baselines, improving AUPR by 7.0% on MFNP and 10.2% on QENP. (2) Interestingly, the simple LogReg baseline is often competitive, and in some cases even outperforms deep learning methods. We attribute this to the small and noisy nature of the data, where simpler models are less prone to overfitting. This highlights the need for caution and careful design when applying deep learning models in this domain. (3) Diffusion models struggle with training instability and high data needs; our composite flow matching improves both stability and data efficiency.

Table 2 shows consistent drops when removing either component, confirming that the composite base improves data efficiency and the detection head addresses imperfect observation.

We further analyze where WILDFLOW works particularly well. For each cell i and month t , we compute the log-loss difference relative to LogReg. Smaller values indicate regions where WILDFLOW provides more accurate and calibrated probabilistic forecasts. In Figure 3, we observe that WILDFLOW consistently achieves stronger performance on cells with higher levels of adjacent patrol effort across all test years. Neighboring patrol effort can induce displacement effects and create intricate spatial dependencies, and the superior performance of our method in these regions supports the intuition that flow matching is well-suited to modeling complex, high-dimensional spatial patterns. We also find that WILDFLOW tends to perform better on cells located closer to patrol posts. A possible explanation is that proximity to patrol posts increases both the frequency and reliability of detections, which our method is better able to exploit.

6 Conclusion and Path to Deployment

We introduced WILDFLOW, a generative AI framework for poaching prediction that addresses observation bias and data scarcity. On datasets from two Ugandan parks, it outperforms strong baselines, highlighting its potential for conservation and climate resilience. As next steps, we will run pilots in Uganda and Nigeria with wildlife authorities, integrating the tool into SMART software and refining it through ranger feedback. Our ultimate goal is large-scale, responsible deployment of generative AI to support biodiversity protection under climate change.

Acknowledgement

We are thankful to the Uganda Wildlife Authority for granting us access to incident data from Murchison Falls and Queen Elizabeth National Park. We also thank the anonymous reviewers for their valuable feedback. This work was supported by ONR MURI N00014-24-1-2742.

References

- [1] Sarah R Weiskopf, Forest Isbell, Maria Isabel Arce-Plata, Moreno Di Marco, Mike Harfoot, Justin Johnson, Susannah B Lerman, Brian W Miller, Toni Lyn Morelli, Akira S Mori, et al. Biodiversity loss reduces global terrestrial carbon storage. *Nature Communications*, 15(1):4354, 2024.
- [2] George Wittemyer, Joseph M Northrup, Julian Blanc, Iain Douglas-Hamilton, Patrick Omondi, and Kenneth P Burnham. Illegal killing for ivory drives global decline in african elephants. *Proceedings of the National Academy of Sciences*, 111(36):13117–13121, 2014.
- [3] World wildlife crime report 2020: Pangolins. Technical report, United Nations Office on Drugs and Crime, Vienna, 2020. URL https://www.unodc.org/documents/wwcr/2020/Wildlife_crime_Pangolin_UNODC.pdf.
- [4] KT Everatt, R Kokes, and C Lopez Pereira. Evidence of a further emerging threat to lion conservation; targeted poaching for body parts. *Biodiversity and Conservation*, 28(14):4099–4114, 2019.
- [5] Lily Xu, Andrew Perrault, Fei Fang, Haipeng Chen, and Milind Tambe. Robust reinforcement learning under minimax regret for green security. In *Uncertainty in Artificial Intelligence*, pages 257–267. PMLR, 2021.
- [6] Lauric Thiault, Damian Weekers, Matt Curnock, Nadine Marshall, Petina L Pert, Roger Beeden, Michelle Dyer, and Joachim Claudet. Predicting poaching risk in marine protected areas for improved patrol efficiency. *Journal of Environmental Management*, 254:109808, 2020.
- [7] Debarun Kar, Benjamin Ford, Shahrzad Gholami, Fei Fang, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, et al. Cloudy with a chance of poaching: Adversary behavior modeling and forecasting with real-world poaching data. 2017.
- [8] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.
- [9] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2023. URL <https://openreview.net/forum?id=PqvMRDCJT9t>.
- [10] Aidan Keane, Julia PG Jones, and EJ Milner-Gulland. Encounter data in resource management and ecology: pitfalls and possibilities. *Journal of Applied Ecology*, 48(5):1164–1173, 2011.
- [11] Darryl I MacKenzie, James D Nichols, Gideon B Lachman, Sam Droege, J Andrew Royle, and Catherine A Langtimm. Estimating site occupancy rates when detection probabilities are less than one. *Ecology*, 83(8):2248–2255, 2002.
- [12] Rong Yang, Benjamin Ford, Milind Tambe, and Andrew Lemieux. Adaptive resource allocation for wildlife protection against illegal poachers. In *Proceedings of the 2014 international conference on Autonomous agents and multi-agent systems*, pages 453–460, 2014.
- [13] Thanh H Nguyen, Arunesh Sinha, Shahrzad Gholami, Andrew Plumptre, Lucas Joppa, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, and Rob Critchlow. Capture: A new predictive anti-poaching tool for wildlife protection. 2016.

- [14] Shahrzad Gholami, Benjamin Ford, Fei Fang, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, and Joshua Mabonga. Taking it for a test drive: a hybrid spatio-temporal model for wildlife poaching prediction evaluated through a controlled field test. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 292–304. Springer, 2017.
- [15] Shahrzad Gholami, Sara Mc Carthy, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Margaret Driciru, Fred Wanyama, Aggrey Rwetsiba, Mustapha Nsubaga, Joshua Mabonga, et al. Adversary models account for imperfect crime data: Forecasting and planning against real-world poachers. *International Conference on Autonomous Agents and Multiagent Systems*, 2018.
- [16] Lily Xu, Shahrzad Gholami, Sara McCarthy, Bistra Dilkina, Andrew Plumptre, Milind Tambe, Rohit Singh, Mustapha Nsubaga, Joshua Mabonga, Margaret Driciru, et al. Stay ahead of poachers: Illegal wildlife poaching prediction and patrol planning under uncertainty with field test evaluations (short version). In *2020 IEEE 36th international conference on data engineering (ICDE)*, pages 1898–1901. IEEE, 2020.
- [17] SMART. Spatial monitoring and reporting tool (smart). <http://smartconservationtools.org/>, 2013. Accessed: 2025-08-19.
- [18] Rob Critchlow, Andrew J Plumptre, Margaret Driciru, Aggrey Rwetsiba, Emma J Stokes, Charles Tumwesigye, Fred Wanyama, and CM Beale. Spatiotemporal trends of illegal activities from ranger-collected data in a ugandan national park. *Conservation biology*, 29(5):1458–1470, 2015.
- [19] Lingkai Kong, Haichuan Wang, Yuqi Pan, Cheol Woo Kim, Mingxiao Song, Alayna Nguyen, Tonghan Wang, Haifeng Xu, and Milind Tambe. Robust optimization with diffusion models for green security. *arXiv preprint arXiv:2503.05730*, 2025.
- [20] Renwu Gao, Siting Zheng, Jia He, and Linlin Shen. CycleGAN-based image translation for near-infrared camera-trap image recognition. In *International Conference on Pattern Recognition and Artificial Intelligence*, pages 453–464. Springer, 2020.
- [21] Qiaoyi Zhang, Xiaoli Yi, Jiali Guo, Yadong Tang, Tao Feng, and Rui Liu. A few-shot rare wildlife image classification method based on style migration data augmentation. *Ecological Informatics*, 77:102237, 2023.
- [22] Eunbeen Kim, Jaeuk Moon, Jonghwa Shim, and Eenjun Hwang. DualDiscWaveGAN-based data augmentation scheme for animal sound classification. *Sensors*, 23(4):2024, 2023.
- [23] Anthony Gibbons, Emma King, Ian Donohue, and Andrew Parnell. Generative ai-based data augmentation for improved bioacoustic classification in noisy environments. *arXiv preprint arXiv:2412.01530*, 2024.
- [24] José Sebastián Ñungo Manrique, Francisco Gómez, and Freddy Hernández-Romero. Anuran call synthesis with diffusion models for enhanced bioacoustic classification under data scarcity. *Ecological Informatics*, page 103322, 2025.
- [25] Juliana Silva Barbosa, Ulhas Gondhali, Gohar Petrossian, Kinshuk Sharma, Sunandan Chakraborty, Jennifer Jacquet, and Juliana Freire. A cost-effective llm-based approach to identify wildlife trafficking in online marketplaces. *Proceedings of the ACM on Management of Data*, 3(3):1–23, 2025.
- [26] Lingkai Kong, Haichuan Wang, Tonghan Wang, Guojun Xiong, and Milind Tambe. Composite flow matching for reinforcement learning with shifted-dynamics data. *arXiv preprint arXiv:2505.23062*, 2025.
- [27] Pauli Virtanen, Ralf Gommers, Travis E. Oliphant, Matt Haberland, Tyler Reddy, David Cournapeau, Evgeni Burovski, Pearu Peterson, Warren Weckesser, Jonathan Bright, Stéfan J. van der Walt, Matthew Brett, Joshua Wilson, K. Jarrod Millman, Nikolay Mayorov, Andrew R. J. Nelson, Eric Jones, Robert Kern, Eric Larson, C J Carey, İlhan Polat, Yu Feng, Eric W. Moore, Jake VanderPlas, Denis Laxalde, Josef Perktold, Robert Cimrman, Ian Henriksen, E. A. Quintero, Charles R. Harris, Anne M. Archibald, Antônio H. Ribeiro, Fabian Pedregosa, Paul

- van Mulbregt, and SciPy 1.0 Contributors. SciPy 1.0: Fundamental Algorithms for Scientific Computing in Python. *Nature Methods*, 17:261–272, 2020. doi: 10.1038/s41592-019-0686-2.
- [28] Jacob Gardner, Geoff Pleiss, Kilian Q Weinberger, David Bindel, and Andrew G Wilson. Gpytorch: Blackbox matrix-matrix gaussian process inference with gpu acceleration. *Advances in neural information processing systems*, 31, 2018.
- [29] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *International Conference on Learning Representations*, 2017. URL <https://openreview.net/forum?id=SJU4ayYgl>.
- [30] Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *International Conference on Learning Representations (ICLR)*, 2019.

A Appendix

A.1 Extended Related Works

AI for poaching prediction. Yang et al. [12] introduced the Protection Assistant for Wildlife Security (PAWS) using a subjective-utility quantal response (SUQR) behavioral model. To explicitly account for imperfect detection, Nguyen et al. [13] proposed a two-layer Bayesian network with latent variables. However, Kar et al. [7] later found this approach brittle in practice due to model complexity; an ensemble of decision trees achieved higher accuracy with lower runtime and was validated in a one-month field test. To combine the strengths of prior methods, Gholami et al. [14] proposed a geo-clustering technique that yields a hybrid of multiple Markov random fields with a bagging ensemble of decision trees, supported by a five-month controlled field test. To further address nonuniform uncertainty stemming from uneven patrol effort, Gholami et al. [15], Xu et al. [16] trained ensembles of weak learners—decision trees and Gaussian processes—on data stratified by patrol intensity, with deployment via the SMART conservation platform [17]. Finally, Xu et al. [5] used logistic regression to quantify the deterrence effects of ranger patrols on poaching risk.

Despite these practical successes, most approaches still rely on classical models with limited capacity to capture high-dimensional, nonlinear spatial patterns; few make use of modern deep learning. Most recently, Kong et al. [19] applied diffusion models to green security problems. Our work differs in three key ways: (i) Their focus is on robust patrol optimization for general green security, whereas we focus specifically on poaching risk forecasting for conservation. (ii) They do not account for imperfect detection, which is a critical challenge in conservation settings, while we explicitly model the visit-level detection process. (iii) Their features are restricted to historical patrol effort, whereas we incorporate rich environmental covariates. In this richer feature space, diffusion models struggle due to their data-hungry nature. To address this, we propose a composite flow model that improves data efficiency.

Generative AI for wildlife conservation In camera-trap vision, neural generative models have been applied directly to wildlife datasets: CycleGAN variants translate between sensor domains (e.g., visible and near-infrared) and augment rare species to improve few-shot classification [20, 21]. Beyond imagery, generative audio models have been developed to synthesize wildlife vocalizations and strengthen bioacoustic monitoring under data scarcity. Early work explored class-conditional GANs for animal-sound augmentation [22], followed by diffusion-based pipelines that generate birdsong spectrograms to improve classifier accuracy [23], and more recently, diffusion models that synthesize anuran calls [24]. Large language models (LLMs) have also been applied to detect illegal wildlife trafficking on online marketplaces by generating pseudo-labels for unlabeled listings [25]. In contrast, our work focuses on generative modeling for poaching prediction in protected areas.

A.2 Derivation of the Marginalized Log-Likelihood

For cell-month (i, t) , let $r_{i,t} = \sigma(\psi_{i,t})$ and $\mathbf{y}_{i,t} = (y_{i,t,1}, \dots, y_{i,t,J_{i,t}})$ with detection probabilities $\{p_{i,t,j}\}_{j=1}^{J_{i,t}}$. Assume: (i) $z_{i,t} \sim \text{Bernoulli}(r_{i,t})$; (ii) visits are conditionally independent given $z_{i,t}$; (iii) no false positives: $P(y_{i,t,j} = 1 \mid z_{i,t} = 0) = 0$.

We marginalize over $z_{i,t} \in \{0, 1\}$:

$$P(\mathbf{y}_{i,t} \mid \psi_{i,t}, \{p_{i,t,j}\}) = (1 - r_{i,t}) P(\mathbf{y}_{i,t} \mid z=0) + r_{i,t} P(\mathbf{y}_{i,t} \mid z=1). \quad (2)$$

By (iii),

$$P(\mathbf{y}_{i,t} \mid z=0) = \mathbf{1}\{\forall j : y_{i,t,j} = 0\},$$

and by (ii),

$$P(\mathbf{y}_{i,t} \mid z=1) = \prod_{j=1}^{J_{i,t}} p_{i,t,j}^{y_{i,t,j}} (1 - p_{i,t,j})^{1-y_{i,t,j}}.$$

Substituting into (2) yields

$$P(\mathbf{y}_{i,t} \mid \psi_{i,t}, \{p_{i,t,j}\}) = (1 - r_{i,t}) \mathbf{1}\{\forall j : y_{i,t,j} = 0\} + r_{i,t} \prod_{j=1}^{J_{i,t}} p_{i,t,j}^{y_{i,t,j}} (1 - p_{i,t,j})^{1-y_{i,t,j}}.$$

Define

$$S_{i,t} = \mathbf{1}\{\exists j : y_{i,t,j} = 1\}, \quad L_{i,t} = \prod_{j=1}^{J_{i,t}} p_{i,t,j}^{y_{i,t,j}} (1 - p_{i,t,j})^{1-y_{i,t,j}}.$$

Since $\mathbf{1}\{\forall j : y_{i,t,j} = 0\} = 1 - S_{i,t}$, the log-likelihood term is

$$\log P(\mathbf{y}_{i,t} \mid \cdot) = \log((1 - r_{i,t})(1 - S_{i,t}) + r_{i,t} L_{i,t}),$$

and the training loss is its negative.

Special cases. If $S_{i,t} = 0$ (no detections),

$$\log P(\mathbf{y}_{i,t} \mid \cdot) = \log((1 - r_{i,t}) + r_{i,t} \prod_j (1 - p_{i,t,j})).$$

If $S_{i,t} = 1$ (at least one detection),

$$\begin{aligned} \log P(\mathbf{y}_{i,t} \mid \cdot) &= \log r_{i,t} \\ &+ \sum_j \left(y_{i,t,j} \log p_{i,t,j} \right. \\ &\quad \left. + (1 - y_{i,t,j}) \log(1 - p_{i,t,j}) \right). \end{aligned}$$

A.3 Training Algorithm

We provide the full training algorithm in Algorithm 1.

Stage 1: Training encoder and detector. In Stage 1, we jointly train two components: (i) an encoder that estimates latent occupancy logits, and (ii) a detection head that models visit-level detection probabilities.

For each cell i at month t , the node feature is defined as

$$\mathbf{c}'_{i,t} = [\mathbf{x}_{i,t}, S_{i,t}, a_{i,t}^m],$$

where $\mathbf{x}_{i,t}$ are geographic covariates, $S_{i,t} = \mathbf{1}\{\exists j : y_{i,t,j} = 1\}$ indicates whether poaching was observed, and $a_{i,t}^m = \sum_j a_{i,t,j}$ is the total patrol effort that month. Let $\mathbf{C}'_t$ denote the collection of all node features in month t .

The encoder maps the graph and node features to latent occupancy logits:

$$\hat{\psi}_t^{(1)} = f_\omega(G, \mathbf{C}'_t).$$

Independently, the detection head computes the probability of detecting a snare on visit j as

$$p_{i,t,j} = \sigma(g_\phi(\mathbf{x}_{i,t}, a_{i,t,j})),$$

which depends only on visit-level covariates and patrol effort.

Substituting $r_{i,t} = \{\sigma(\hat{\psi}_{i,t})\}$ and $\{p_{i,t,j}\}$ into Eq. (1) yields the occupancy–detection log-likelihood $\mathcal{L}_{\text{occ}}(i, t)$. We then jointly optimize encoder and detector parameters (ω, ϕ) by solving

$$\max_{\omega, \phi} \sum_{i,t} \mathcal{L}_{\text{occ}}(i, t).$$

Stage 2: Training latent flow matching. In Stage 2, we train a conditional flow prior to transport a composite base distribution to the latent targets inferred by the Stage 1 encoder. At this stage, the encoder and detector parameters (ω, ϕ) are frozen, and the encoder outputs $\hat{\psi}_t^{(1)}$ serve as the training targets.

We train a graph–conditional velocity field $v_\theta(\cdot, \cdot; G, \mathbf{C}_t)$ using the composite base $\psi_t^{(0)} = b_\eta(\mathbf{C}_t) + \epsilon$ with $\epsilon \sim \mathcal{N}(\mathbf{0}, \sigma_0^2 \mathbf{I})$, and straight-line paths $\psi_t^{(s)} = (1 - s)\psi_t^{(0)} + s\hat{\psi}_t^{(1)}$, $s \in [0, 1]$. Prior work has shown that when the base distribution is closer to the target distribution, the generalization performance of flow matching improves [26].

Putting the components together, the conditional flow-matching objective is defined as

$$\min_{\theta} \sum_t \mathbb{E}_{\substack{s \sim \mathcal{U}[0,1] \\ \epsilon \sim \mathcal{N}(\mathbf{0}, \sigma_0^2 \mathbf{I})}} \left\| v_\theta(\psi_t^{(s)}, s; G, \mathbf{C}_t) - (\hat{\psi}_t^{(1)} - \psi_t^{(0)}) \right\|_2^2$$

Inference. For a forecast month t^* , we construct $\mathbf{C}_{t^*}$ from features and *past* patrol effort only. We then sample an initial base $\psi_{t^*}^{(0)} \sim p_0(\cdot | \mathbf{C}_{t^*})$, and integrate the learned velocity field

$$\frac{d\psi_{t^*}^{(s)}}{ds} = v_\theta(\psi_{t^*}^{(s)}, s; G, \mathbf{C}_{t^*}), \quad s \in [0, 1],$$

to obtain $\psi_{t^*}^{(1)}$. The resulting occupancy risk is $\mathbf{r}_{t^*} = \sigma(\tilde{\psi}_{t^*})$. For more stable estimates, we can draw M samples and compute the Monte Carlo mean.

Given planned visit-level efforts $\{a_{i,t^*,j}\}_{j=1}^{J_{i,t^*}}$, detection probabilities are obtained from the trained head:

$$p_{i,t^*,j} = \sigma(g_\phi(\mathbf{x}_{i,t^*}, a_{i,t^*,j})).$$

The probability of *at least one* detection in cell i during month t^* is then

$$\hat{p}_{\text{any},i,t^*} = r_{i,t^*} \left(1 - \prod_{j=1}^{J_{i,t^*}} (1 - p_{i,t^*,j}) \right).$$

Note that the encoder is only used during training and is not required at inference time.

A.4 Experimental Details

LogReg: The occupancy model is linear, and the detection component is modeled with a linear head, consistent with the other baselines. Predictions are made independently for each cell. To capture spatial spillovers from patrol activity, we follow Xu et al. [5] and include aggregated patrol effort in adjacent cells as an additional feature in the occupancy model. We jointly train the linear occupancy model and detection head by minimizing $-\sum_{i,t} \mathcal{L}_{\text{occ}}(i, t)$, where $\mathcal{B}$ denotes the batch. Optimization is performed using the BFGS algorithm from SciPy [27].

MLP: We implement a multilayer perceptron (MLP) baseline for occupancy modeling, using a three-layer fully connected network with ReLU activations and dropout regularization. To ensure numerical stability, the output logits are clipped to an absolute value of 10 before applying the sigmoid transformation. Predictions are made independently for each cell. To capture spatial spillovers from patrol activity, we follow Xu et al. [5] and include aggregated patrol effort in adjacent cells as an additional feature in the occupancy model. The model is trained by minimizing $-\sum_{i,t} \mathcal{L}_{\text{occ}}(i, t)$. The MLP occupancy network and detection head are optimized jointly using the AdamW optimizer, with hyperparameters listed in Table 3.

Algorithm 1: Two-Stage Training for WILDFLOW

Input: Monthly graphs $\{G_t = (V, E)\}$ with node features $\mathbf{C}_t$ and $\mathbf{C}'_t$; visit-level data $\{(a_{i,t,j}, y_{i,t,j})\}$; encoder f_ω , detection head g_ϕ , flow velocity v_θ ; pretrained linear occupancy model b_η .

Output: Trained $f_\omega, g_\phi, v_\theta$.

```
1 Stage 1:
2 Encoder & Detection Head
3 foreach  $epoch = 1, \dots, E_1$  do
4   foreach  $month\ t$  do
5      $\hat{\psi}_t \leftarrow f_\omega(G_t, \mathbf{C}'_t)$  ▷ logits for latent occupancy
6     foreach  $cell\ i \in V$  and  $visit\ j = 1, \dots, J_{i,t}$  do
7        $p_{i,t,j} \leftarrow \sigma(g_\phi(\mathbf{x}_{i,t}, a_{i,t,j}))$  ▷ visit-level detection prob.
8        $\mathcal{L}_{occ}(t) \leftarrow \sum_i \log p(\{y_{i,t,j}\}_j | \hat{\psi}_{i,t}, \{p_{i,t,j}\}_j)$  ▷ Eq. (1)
9    $\text{Update } (\omega, \phi) \text{ by maximizing } \sum_t \mathcal{L}_{occ}(t)$ 
10   $\text{▷ Outputs surrogate targets } \hat{\psi}_t^{(1)} \leftarrow f_\omega(G_t, \mathbf{C}'_t).$ 
11 Stage 2:
12 Latent Flow Matching
13 Freeze  $(\omega, \phi)$ 
14 foreach  $epoch = 1, \dots, E_2$  do
15   foreach  $month\ t$  do
16     Sample base  $\psi_t^{(0)} \sim \mathcal{N}(b_\eta(\mathbf{C}_t), \sigma_0^2 I)$ 
17     Draw  $s \sim \mathcal{U}[0, 1]$  and form  $\psi_t^{(s)} = (1-s)\psi_t^{(0)} + s\hat{\psi}_t^{(1)}$ 
18     Compute target velocity  $u_t = \hat{\psi}_t^{(1)} - \psi_t^{(0)}$ 
19     Predict  $v_t = v_\theta(\psi_t^{(s)}, s; G_t, \mathbf{C}_t)$ 
20      $\mathcal{L}_{FM}(t) \leftarrow \|v_t - u_t\|_2^2$ 
21    $\text{Update } \theta \text{ by minimizing } \sum_t \mathcal{L}_{FM}(t)$ 
22 return  $(f_\omega, g_\phi, v_\theta)$ 
```

GP: We model occupancy using a sparse variational Gaussian process (GP), where the latent function f is mapped to occupancy probability via a sigmoid transformation. Predictions are made independently for each cell. To capture spatial spillovers from patrol activity, we follow Xu et al. [5] and include aggregated patrol effort in adjacent cells as an additional feature in the occupancy model. The GP and linear detection head are trained jointly by maximizing the evidence lower bound (ELBO):

$$\mathcal{L}_{GP} = -\mathbb{E}_{q(f)}[\log p(y | f)] + \text{KL}(q(f) \| p(f)),$$

where $p(f)$ is the GP prior and $q(f)$ is a sparse variational posterior defined using inducing points to approximate the true posterior. The expectation is estimated via Monte Carlo sampling. We use the GPyTorch implementation [28], and the detailed hyperparameter setup is provided in Table 3.

GNN: We represent each cell as a node in a graph, with edges defined by spatial adjacency. Node features and graph structure are passed through a GCN encoder [29], which outputs an occupancy probability for each node via a sigmoid transformation. The encoder and detection head are trained jointly by minimizing $-\sum_{i,t} \mathcal{L}_{occ}(i, t)$. We optimize using AdamW [30] and apply gradient clipping with a maximum norm of 5. Hyperparameter details are provided in Table 3.

Transformer: As with the GNN, we represent each cell as a node in a graph, with edges defined by spatial adjacency. Node features and graph structure are passed through a Transformer-based graph encoder, which performs multi-head attention over each node’s neighborhood and outputs an occupancy probability via a sigmoid transformation. The encoder and detection head are trained jointly by minimizing $-\sum_{i,t} \mathcal{L}_{occ}(i, t)$, using AdamW [30] with gradient clipping at a maximum norm of 5. Hyperparameter details are provided in Table 3.

Diffusion Model: We adopt the same two-stage training procedure as WILDFLOW. Hyperparameter settings are listed in Table 3.

WILDFLOW: The hyperparameter settings for our method are also summarized in Table 3.

Model	HP	Value
MLP	Epochs	100
	LR	$\{3, 5, 8\} \times 10^{-3}$
	Optimizer	AdamW
	Batch	512
	W. Decay	10^{-4}
	Logit Bound	10
GP	Epochs	100
	LR	$\{3, 5\} \times 10^{-3}$
	Optimizer	AdamW
	Batch	256
	MC Samples	8 / 50
GNN	Epochs	120
	LR	$\{3, 5\} \times 10^{-3}$
	Optimizer	AdamW
	Batch	256
	Hidden Dim	128
	Layers	2
Transformer	Epochs	120
	LR	$\{3, 5\} \times 10^{-3}$
	Optimizer	AdamW
	Batch	256
	Hidden Dim	128
	Layers	2
Diffusion (Stage I)	Heads	4
	Optimizer	AdamW
	LR	10^{-2}
	Batch	256
	Hidden Dim	128
Diffusion (Stage II)	Layers	2
	Optimizer	AdamW
	LR	$\{10^{-2}, 10^{-3}\}$
	Batch	256
	Hidden Dim	128
WILDFlow (Stage I)	Layers	2
	Optimizer	AdamW
	LR	10^{-2}
	Batch	256
	Hidden Dim	128
WILDFlow (Stage II)	Layers	2
	Optimizer	AdamW
	LR	$\{10^{-2}, 10^{-3}\}$
	Batch	256
	Hidden Dim	128
	σ_0	0.1

Table 3: Hyperparameter settings for all methods.